AMENDMENTS

BRIEF SUMMARY OF THE INVENTION

Please amend the Brief Summary of the Invention section as follows:

Please replace paragraph [0011] with the following amended paragraph:

[0011] In one embodiment, a method of identifying a type of encoding used in generating a voice data stream incorporates reading words of the voice data stream, determining a first number of words of the voice data stream that corresponds to a first range of values, determining a second number of words of the voice data stream that corresponds to a second range of values, generating µ-law linear equivalents of the one or more words of the voice data stream, determining a third number of words corresponding to the µ~~m~~-law linear equivalents of the one or more words that have values within a third range, determining a fourth number of words corresponding to the µ~~m~~-law linear equivalents of the one or more words that have values within a fourth range, generating A-law linear equivalents of the one or more words of the voice data stream, determining a fifth number of words using corresponding to the A-law linear equivalents of the one or more words that have values within a fifth range, and determining a sixth number of words corresponding to the A-law linear equivalents of the one or more words that have values within a sixth range.

DETAILED DESCRIPTION OF THE INVENTION

Please amend the Detailed Description of the Invention as follows:

Please replace paragraph [0019] with the following amended paragraph:

[0019] Aspects of the present invention may be found in a system and method to detect or identify one or more types of algorithms used in the encoding of a voice or speech waveform. The system and method may be used as a testing tool to identify whether a voice data stream is encoded using one or more pulse code modulation (PCM) compression algorithms defined by ITU (International Telecommunications Union) G.711 recommendation specification. The

system and method may be applied to a voice data stream comprising a number of bytes of data that has been previously stored as a data file. The one or more types of algorithms may comprise a 16 bit linear (in some instances described as uniform PCM or linear G.711), µ-law G.711, and A-law G.711 types of pulse code modulation (PCM) algorithms. The system and method characterize the voice data stream in terms of one or more parameters that correlate with linear G.711, µ~~m~~-law G.711, or A-law G.711. Thereafter, the parameters are analyzed by way of one or more tests to determine which algorithm was used to encode the voice data stream.

Please replace paragraph [0025] with the following amended paragraph:

[0025] Referring to Figure 3A, at step 304, the G.711 detection software operates on a voice data stream file. The file may comprise voice data encoded in linear G.711, µ-law G.711, or A-law G.711. The file may comprise, for example, a size of 800 kilobytes, lasting approximately 100 seconds of audio runtime. At step 308, all overflows and zeros counters are reset to zero. There are two pairs of overflows/zeros counters that are used in associating words that correspond to "zeros" or "overflows" during a µ-law to linear conversion or an A-law to linear conversion. Next at step 312, both µ-law and A-law maximum discontinuity jump registers are set to zero. As was described in Figures 2A and 2B, a maximum discontinuity jump register (MDJR) is used to determine the largest difference between successive linear equivalent values over the entire voice data stream or voice data stream file. Thereafter, at step 316, the word counter is set to zero. In this embodiment, each word or data sample is defined as one byte, in which one byte comprises eight binary digits. At step 320, a word from the data stream is read and converted to its µ-law and A-law linear equivalents. Next, at step 324, the word counter is incremented by one. Now referring to Figure 3B, a histogram of hexadecimal words may be generated based on the values read. In this embodiment, the value of an exemplary 8 bit µ-law or A-law hexadecimal word corresponds to one of 256 intervals within the histogram. The number of bits used to represent an element of the histogram may be proportional to the number of data words comprising the voice data stream file. For example, 32 bits (corresponding to a maximum count of 232) may be used to sufficiently represent an 800 kilobyte (or in this instance an 800 kiloword) voice data stream file. The 256 different hexadecimal values implement 256 x-axis intervals in an exemplary histogram, while the frequency of occurrence of a particular value is

indicated on the y-axis of the histogram by way of the 32-bit counter. Hence, at step 328, the appropriate intervals in the histogram are updated in terms of their occurrence. At step 332, the corresponding µ-law or A-law overflows counters are incremented if the word values exceed their respective thresholds. Optionally, the corresponding µ-law or A-law zeros counters may be incremented if the linear equivalents are below their respective thresholds. Alternatively, the number of words with linear equivalents corresponding to overflows or zeros values may be determined by summing portions of the histogram corresponding to their appropriate µ~~m~~-law or A-law linear equivalents (as will be described in Figure 4 with respect to the calculation of the number of zeros). Next at step 336, the µ-law DJR, is updated, if necessary, by calculating the difference between the µ~~m~~-law linear equivalent value of the word currently read and the µ~~m~~-law linear equivalent value of the word previously read. If this difference is greater than what is currently stored in the µ-law DJR, the difference is used to replace the value currently stored in the µ~~m~~-law MDJR. Hence, after all words in a voice data stream are evaluated by the G.711 detection system, the largest difference between successive word values is stored in the µ~~m~~-law MDJR. Similarly, the A-law MDJR, is updated, if necessary, by calculating the difference between the A-law linear equivalent value of the word currently read and the A-law linear equivalent value of the word previously read. At step 340, the process ends if the entire voice data stream has been read. Otherwise the process advances to step 344. At this step, the process reverts back to step 320, allowing another word to be read from the voice data stream.

Please replace paragraph [0026] with the following amended paragraph:

[0026] Figure 4 is an operational flow diagram illustrating the calculation of a number of parameters which are used in determining the type of G.711 encoding represented by the voice data stream file. At step 404, µ~~m~~-law or A-law words whose linear equivalents correspond to "zeros" (termed µ-law or A-law zeros, hereinafter) may be determined by identifying the corresponding intervals in the histogram. For example, the hexadecimal values - 0x7f, 0xff, 0x7e, and 0xfe may be identified as one or more intervals in the histogram that correspond to µ~~m~~-law zeros. Adding the occurrences represented by these law zero" intervals yields the total number of µ~~m~~-law words in the voice data stream that correspond to "µ~~m~~-law zeros". Likewise, the hexadecimal values - 0x55, 0xd5, 0x54, and 0xd4 may be used to identify appropriate

intervals in the histogram corresponding to A-law zeros. Adding the occurrences represented by these "A-law zero" intervals yields the number of A-law words in the voice data stream that correspond to "A-law zeros". Although previously described and implemented in Figures 3A and 3B using counters, it is contemplated that µ~~m~~-law or A-law words whose linear equivalents correspond to "overflows" (termed µ-law or A-law overflows, hereinafter) may be determined by identifying the appropriate intervals in the histogram and summing the occurrences. Next, at step 408, the corresponding percentages are calculated for linear G.711, µ~~m~~-law G.711 and A-law G.711 zeros. For example, the percentage of linear zeros is calculated by dividing the number of "linear zeros" by the total number of words in the data stream file and then multiplying by 100. Likewise, the percentage of µ~~m~~-law G.711 zeros is calculated in a similar fashion. Similarly, the percentage of A-law G.711 zeros is calculated. Next, at step 412, the percentages are calculated for the number of linear G.711, µ~~m~~-law G.711, and A-law G.711 overflows determined previously.

Please replace paragraph [0027] with the following amended paragraph:

[0027] Thereafter, at step 416, the normalized sum of µ~~m~~-law and A-law "zeros" are calculated using the following equation:

zero_mag = (azero_percent + µzero_percent)/100.0, wherein

zero_mag is defined as the normalized sum of µ-law and A-law zeros;

azero_percent is defined as the percentage of words at A-law zero levels (whose absolute value is below a threshold), and

mzero_percent is defined as the percentage of words at µ~~m~~-law zero levels (whose absolute value is above a threshold).

Please replace paragraph [0028] with the following amended paragraph:

[0028] Next, at step 420, the normalized sum of µ~~m~~-law and A-law "overflows" are calculated using the following equation:

ovfl_mag = (aovfl_percent + movfl_percent)/100.0, wherein

ovfl_mag is defined as the normalized sum of µ-law and A-law overflows;

aovfl_percent is defined as the percentage of words at A-law overflow levels (whose absolute value is above a threshold); and

movfl_percent is defined as the percentage of words at µ~~m~~-law overflow levels (whose absolute value is below a threshold).

Please replace paragraph [0029] with the following amended paragraph:

[0029] Thereafter, at step 424, the normalized difference between µ~~m~~-law and A-law "zeros" are calculated, using the following exemplary equation:

zero_diff = (abs(azero_percent - µzero_percent)/(azero_percent + µzero_percent + 0.001)), wherein

zero_diff is defined as the normalized difference between µ-law and alaw zeros;

µzero_percent is defined as the percentage of words at µ-law zero levels (as was previously described); and

azero_percent is defined as the percentage of words at A-law zero levels (as was previously described).

Please replace paragraph [0031] with the following amended paragraph:

[0031] At the last step 428, of Figure 4, the normalized sums of µ~~m~~-law and A-law "overflows" are calculated using the following equation:

ovfl_diff = (abs(µovfl_percent - aovfl_percent)/(µovfl_percent + aovfl_percent + 0.001)), wherein,

ovfl_diff is defined as the normalized difference between µ-law and A-law overflows;

µovfl_percent is defined as the percentage of words at µ-law overflow levels; and

aovfl_percent is defined as the percentage of words at A-law overflow levels.

Please replace paragraph [0034] with the following amended paragraph:

[0034] Referring to Figure 5, at step 504, the G.711 detection software initiates the start of a new testing sequence by setting N=1. The variable N is an indicator of which test is being executed by the G.711 detection software. At step 508, the first test (N=1, Test #1) is performed. During the course of the first test, a number of decisions are made by the first test based on one or more

parameters calculated previously. For example, at step 512, the first test may determine whether the voice data stream file being tested represents linear G.711 file. If the test determines that the voice data stream is linear G.711, it returns an appropriate message such as "Return Linear G.711". At step 516, the first test may determine whether the voice data stream file represents µ~~m~~-law G.711 file. If the test determines that the voice data stream is µ~~m~~-law G.711, it returns an appropriate message. At step 520, the first test may determine whether the voice data stream represents an A-law G.711 file. Next, at step 524, the first test may determine that the voice data stream is not characteristic of linear, µ~~m~~-law, or A-law G.711. As a consequence, the first test may generate an "unknown" response. Otherwise, at step 528, the process proceeds to the next test. At step 532, N is incremented by one, so N=2, and the testing process reverts to step 508 with the second test being performed. Similarly, the testing process continues until a decision is made by a test or until the last test is completed. The following ten tests may be performed sequentially to determine the type of G.711 represented by a voice data stream file. The embodiments provided by the following ten tests are exemplary, and it is contemplated that other similar tests may be implemented using the parameters previously determined in Figures 2 through Figure 5.

Please replace paragraph [0035] with the following amended paragraph:

[0035] The first test determines if both a µ-law maximum jump discontinuity and an A-law maximum jump discontinuity are greater than a first threshold. In addition, the test determines if a difference between the µ-law maximum jump discontinuity and a linear maximum jump discontinuity is greater than a second threshold. Furthermore, the test determines if a difference between an A-law maximum jump discontinuity and the linear maximum jump discontinuity is greater than the second threshold. Then, the first test verifies if a normalized sum of µ~~m~~-law and A-law "overflows" is above a third threshold, a percentage of linear overflows is less than a fourth threshold, a percentage of µ-law overflows is greater than a fifth threshold, and a percentage of A-law overflows is greater than the fifth threshold. If all these conditions are satisfied, the G.711 detection software determines that the voice data stream file is linear G.711. For example, a software program such as a C/C++ program may comprise the following high level language instructions to implement this particular test, in which exemplary threshold values

for JUMP_MAX, JUMP_DIFF, THR_LIN_OVFL_PERCENT, THR_OVFL_MAG, THR_UA_OVFL_PERCENT, and THR_UA_OVFL_PERCENT were defined previously.

Please replace paragraph [0047] with the following amended paragraph:

[0047] As illustrated by the preceding output, the samples or words in the voice data stream file are characterized by a substantial number of A-law zeros. The values of these words, after converting from A-law to linear are analyzed and those words that exceed a particular threshold value are categorized as overflows while those that fall below a particular threshold are classified as zeros. In this particular data stream file, the percentage of A-law zeros far exceeds the percentage of μ-law zeros or linear zeros. Referring to the output above, the percentage of A-law zeros is 93.69% while the μ-law and linear zeros are negligible. Another parameter of significance is the maximum discontinuity jump associated with values of successive words in either the linear, A-law, or μ-law case. As illustrated in the output, the maximum discontinuity jump associated with the A-law case is the smallest among the three possible cases. The maximum discontinuity jump associated with A-law is 11,766 compared with approximately 64,000 for the other two cases, indicating that a voice data stream decoded using A-law G.711 results in values that are more reasonable than the same voice data stream decoded using either μ~~m~~-law G.711 or linear G.711. Hence, as illustrated by the last line of the output, the data stream file has been determined to be encoded using A-law (i.e., the data file is a representation of A-law).

CLAIMS

Please amend Claims 1, 9, 20, 28, 30, 33, 36-37, 45 and 52 and add new Claims 57-88 as shown in the Listing of the Claims that follows. This listing replaces any prior listings of claims concerning the present Application.